

REMARKS

Claims 39 and 50 have been canceled. Claims 1-38 and 40-49 are pending in the application upon entry of this amendment. Reconsideration is respectfully requested in light of the following remarks.

37 CFR 1.75(c) Objection:

The Examiner objected to claims 39 and 50 under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. While Applicants do not necessarily agree with this objection, claims 39 and 50 have been cancelled to expedite prosecution.

Double Patenting Rejections:

The Examiner rejected claims 1 and 14 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 and 7 of pending U.S. Patent Application No. 10/087,652. The Examiner also rejected claims 1, 14, 26, 29 and 40 as being unpatentable over claims 1, 7, 15 and 21 of pending U.S. Patent Application No. 10/087,224 and claims 1, 7, 15 and 21 of pending U.S. Patent Application No. 10/087,225. Claims 1, 14, 29 and 40 as being unpatentable over claims 1, 8 and 9 of pending U.S. Patent Application No. 10/087,197. Claim 26 as being unpatentable over claims 1, 10, 17, 26 of pending U.S. Patent Application No. 10/087,234.

Applicants traverse these rejections on the grounds that the Examiner has not stated a proper *prima facie* rejection. According to MPEP 804.II.B.1, “the analysis employed in an obviousness-type double patenting determination parallels the guidelines for a 35 U.S.C. 103(a) rejection.” This section of the MPEP also states that the same “factual inquires ... that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are employed when making an obviousness-type

double patenting analysis.” MPEP 804.II.B.1 also states that the Examiner should list the differences between **each** rejected claim and the claims of the other patent/application, and for **each** difference the Examiner should give the reasons why a person of ordinary skill in the art would conclude that the invention defined in the claim is an obvious variation of the invention defined in a claim of the other patent/application. The Examiner has not addressed all of the specific differences between the claims of the present application and the other cited application. Instead, the Examiner has simply stated broad conclusions of obvious without support of any evidence of record and without address the exact language of the claims.

Nor has the Examiner specifically addressed **each difference** of **each claim** of the present application compared to the claims of the other applications. Instead, the Examiner improperly lumped all the claims together and did not address each specific difference. The Examiner clearly has not met the requirements stated in MPEP 804.II.B.1 to establish a *prima facie* obviousness-type double patenting rejection. Accordingly, Applicants respectfully request removal of the double patenting rejections.

Section 103(a) Rejections:

The Examiner rejected claims 26 and 27 under 35 U.S.C. § 103(a) as being unpatentable over Courts (U.S. Patent 6,360,249) in view of Zaiken (U.S. Patent 5,907,848), claims 29-36, 39-47 and 50 as being unpatentable over Courts in view of Challenger (U.S. Patent 6,026,413), and claims 37, 38, 48 and 49 as being unpatentable over Courts in view of Challenger and in view of Jackson (U.S. Publication 2003/0051145). Applicants respectfully traverse these rejections for at least the following reasons.

Regarding claim 26, contrary to the Examiner's contention, Courts in view of Zaiken fails to teach or suggest means for an application server to track accesses of mutable attributes in a client state of session data. Courts teaches that session data is downloaded from a global session server (220) and provided to a render engine (212) and

that changed data from a session cache is written back to global session data (Courts, col. 8, lines 15-24 and 59-64). However, Courts is silent regarding tracking accesses of mutable attributes in a client state of session data. Zaiken, while not specifically relied on by the Examiner regarding this limitation, also fails to teach or suggest, even if combined with Courts, tracking accesses of mutable attributes.

The Examiner cites col. 9, lines 45-65 and col. 5, lines 20-50 of Courts. However, the Examiner's cited passages do not describe tracking accesses of mutable attributes. Instead, col. 9, lines 45-65 and col. 10, lines 8-25, describe Courts' use of session caches. In particular, Courts describes a session server that provides locked access to session data and session caches. For instance, Courts teaches that a session cache is used to update session data regarding a particular user request and that "[w]hen the web page is built, it is sent to the requesting user" and that "session data 234 is then updated, stored in global session server 232, and released for access by another session cache" (Courts, col. 9, lines 60-63). Thus, this passage describes session caches that first obtain a copy of session data from the session server and then update that copy of the data (e.g., caching). When the user request is completed (e.g., serviced), Courts session cache and stored in global session server. Merely making a copy of session data, modifying it, and then copying it back to a session server does not teach or suggest, even if combined with Zaiken, tracking accesses of mutable attributes.

The Examiner appears to be confusing the mere accessing of mutable attributes with tracking those accesses. Courts describes wholesale copying of updated session data, but does not mention, nor is Courts concerned with, tracking accesses of mutable attributes. Applicants are not arguing that Courts' system does not make accesses of mutable attributes. Applicants are arguing that Courts in view of Zaiken does not teach or suggest tracking accesses of mutable attributes. Thus, while Courts, whether considered singly or in combination, teaches updating session data in a session cache and then storing that updated data to a session server, Courts does not mention anything regarding tracking such accesses of mutable attributes.

The Examiner also cites col. 5, lines 20-50 of Courts, referring to "track to know when the data need to [be] update[d]" (Final Office Action, p. 7). However, this passage describes Courts' load balancing "to evenly distribute requests across any number of web system servers" (Courts, col. 5, lines 21-22). The Examiner argues that together, Courts' monitoring of requests for load balancing and Courts' session caches teaches tracking accesses of mutable attributes. Firstly, Courts' monitoring of requests for load balancing has nothing to do with caching session data or about accesses to mutable attributes in a client state of session data. Monitoring user requests for load balancing involves determining which of several possible servers should service a particular request so that, hopefully, no server will become overloaded and no request will have to wait unduly to be serviced. Courts fails to mention anything about tracking accesses of mutable attributes as part of load balancing or monitoring user requests.

Similarly, Courts' teaching regarding session caches does not include or imply tracking accesses of mutable attributes. Instead, Courts teaches that a session cache within a session manager locks a session ID (SID) associated with a client request and downloads session data for that SID. The cached session data may be updated as part of servicing the user request and then the modified session data from the session cache is copied to (e.g., "stored to") the session server.

The Examiner's reliance on storing updated session data to a global session server is misplaced. As noted above, writing cached (even if modified) data back to a session server storing global session data is not the same as, nor does it teach or suggest, even if combined with Zaiken, tracking accesses of mutable attributes, as recited in Applicants' claim. Simply accumulating a final, modified, version of data in a cache and then copying that modified version of data to a global server does not include the tracking of accesses of mutable attributes.

The Examiner also asserts, "Applicant certainly admitted that Court teaches the claimed limitation", citing pp. 18 and 22 of Applicants' previous response. However, the Examiner is mistaken. Applicant stated, "Courts does not describe tracking accesses of

mutable attributes in the session cache" and "Courts does not mention anything about accesses of mutable attributes as part of dynamic load balancing" (Response to Office Action, September 29, 2005, p. 18).

The Examiner also argues, "Applicant did not claim tracking individual mutable attributes" (Final Office Action, p. 15). However, the Examiner has misunderstood Applicants' argument. Applicants' statement was that Courts does not teach tracking accesses to individual attributes was to illustrate that Courts' caching (and storing) of session data is different from actually tracking accesses of mutable attributes. For instance, the Examiner relies on Courts' "know[ing] when the data need to update" (Final Office Action, p. 7). However, Courts does not mention or teach anything about tracking accesses of mutable attributes in order to "know when the data need to update." Instead, as noted above, Courts teaches storing the updated session data from the session cache to the global session server "[w]hen the web page is built" (Courts, col. 9, lines 60-64). Thus, the Examiner's contention that Courts relies on tracking accesses of mutable attributes to know when the data needs to be updated is clearly incorrect.

Moreover, the Examiner is relying on parts of two different portions of Courts that do not work together or have anything to do with each other. Specifically, the Examiner relies on Courts' monitoring of client requests for load balancing purposes and Courts' session caches. However, Courts' load balancing, and any included monitoring of client requests, does not involve anything about the session caches. The Examiner is attempting to construct Applicants' invention in a piecemeal fashion using various, unrelated functions of Courts' system even though those portions do not, and cannot, function as relied on by the Examiner or as recited in Applicants' claim.

As the Examiner is surely aware, "[i]t is impermissible ... simply to engage in a hindsight reconstruction of the claimed invention, using the applicant's structure as a template and selecting elements from references to fill the gaps." *In re Gorman*, 933 F.2d 982, 987 (Fed. Cir. 1991).

Additionally, Courts in view of Zaiken does not teach or suggest determining a set of mutably accessed attributes of the client state of the session data. The Examiner relies on Courts' render engines as well as Courts session caches. Courts teaches that his render engines can obtain current state information for a session through a session manager and a session cache. Courts' render engines then process requests using the obtained state information (Courts, col. 7, lines 60-65; and col. 10, lines 20-25). However, nowhere does Courts mention determining a set of mutably accessed attributes of the client state of session data. Once again, the Examiner appears to be confusing updating global session data by storing a modified version (from the session cache) to the session server with the specific limitation of determining a set of mutably accessed attributes.

In the Response to Arguments of the Final Action, the Examiner states, "[c]urrent state and session information comprise mutable variable names", citing col. 7, line, 60—col. 8, line 25 and col. 10, lines 8-25. However, the mere presence of mutable variables names or other mutable data does not teach or imply determining a set of mutably accessed attributes, as recited in Applicants' claim. As noted above, Courts teaches that a modified copy of session data in a session cache is simply "stored" to the global session server (col. 9, lines 60-64). The wholesale storing of cached session data does not involve or include determining a set of mutably accessed attributes that are modified with respect to the primary state of the session data.

The Examiner further argues that "[w]hile Courts teaches a subset of modified attributes and the previous version of attributes, the comparison step to get the subset of modified attributes is implied or obvious" (Final. Action, p. 8). This statement is completely unsupported by the evidence of record. The Examiner has failed to show that the comparison step relied on by the Examiner is implied or obvious in view of Courts and Zaiken. The Examiner relies on Zaiken, citing col. 10, lines 1-30. However, the teachings of Zaiken relied on by the Examiner, even if combined with Courts' teachings do not teach or suggest comparing current and previous versions of data to determine a set of mutably accessed attributes of session data. Instead, Zaiken teaches, even if combined with Courts, determining whether a template exists having a user-specified

template ID (Zaiken, col. 9, lines 1-2 and 26-33, and line 65- col. 10, line 11). Specifically, Zaiken teaches that "[t]o make this determination, the program retrieves data fields from the database which correspond to the data values stored in the log record" and that "the data fields are compared to the key value for the template." Zaiken further teaches, "if none of the fields matches the key value, then the template is complete." (Zaiken, col. 10, lines 7-15).

Zaiken is not describing determining a set of mutably accessed attributes, as suggested by the Examiner. Instead, Zaiken teaches searching through log records to determine whether a log record corresponds to an user -specified template ID. Zaiken specifically teaches, "the template ID specified by the user is then compared to the IDs for any existing templates" (col. 9, lines 26-28). The Examiner's cited passage is a further explanation of this process. Searching through log records and comparing them to a user-specified ID is not the same as, nor does it teach or suggest, even if combined with Courts' session caches, determining a set of mutably accessed attributes that are modified with respect to a primary state of session data.

Thus, rather than teaching comparing two versions of data to find a match, as asserted by the Examiner (Final Action, p. 16), Zaiken actually teaches comparing a user specified ID with ID's stored in log records. Zaiken is not comparing a current and a previous version of the data, as relied on by the Examiner and required for the Examiner reasoning to make sense. Instead, Zaiken teaches searching through log records to locate a particular log record based on user-specified search criteria (e.g., user-specified ID).

Furthermore, the Examiner's proposed **combination of Courts and Zaiken would not result in a system including means for an application server to track accesses of mutable attributes in the client state of session data, determine a set of mutably accessed attributes of the client state of the session data of the application server, and determine a subset of the set of mutably accessed attributes that are modified in respect to the primary state of the session data.** Instead, a combination Courts and Zaiken would result only in a system that performs the load balancing and

global session data locking as described in Courts, and includes the template creating ability of Zaiken. Since, as noted above, neither Courts nor Zaiken teaches anything regarding tracking accesses of mutable attributes in a client states of session data, determining a set of mutably accessed attributes, and determining a subset of the set of mutably accessed attributes that are modified in respect to the primary state of session, no combination of Courts and Zaiken would include such functionality.

As shown above, the Examiner's combination of cited art fails to teach or suggest all the limitations of Applicants' claim. As such, the Examiner has failed to provide a *prima facie* rejection. In order to reject a claim as obvious, the Examiner has the burden of establishing a *prima facie* case of obviousness. *In re Warner et al.*, 379 F.2d 1011, 154 U.S.P.Q. 173, 177-178 (C.C.P.A. 1967). To establish a *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP § 2143.03.

For at least the reasons discussed above, the rejection of claim 26 is not supported by the prior art and removal thereof is respectfully requested.

Regarding claim 29, contrary to the Examiner's assertion, **Courts in view of Challenger fails to teach or suggest tracking mutable accesses of a plurality of attributes in a client state of session data**. Instead, Courts teaches only that session data is downloaded from global session server 220 and provided to render engine 212 (*see, e.g.* Courts, column 8, lines 15-24 and lines 59-64) and that changes to a data cache are copied back to the global session data. Courts is silent with regard to *tracking mutable accesses* of the attributes in the client state of the session data.

The Examiner cites column 5, lines 20-50 and column 8, lines 1-25 of Courts. However, neither of the Examiner's cited passages teaches tracking mutable accesses of attributes in a client state of the session data. Instead, the first cited passage (column 5, lines 20-50) describes Courts' use of dynamic load balancing "to evenly distribute requests across any number of web system servers." The second cited passage (column

8, lines 1-25) describes how Courts' caching of session data. Specifically, Courts teaches that a session cache within a session manager calls a "lock" on a SID associated with a client request and downloads session data for that SID from a global session server. After Courts' render engine has built a responsive web page and provided it to the user, the session cache unlocks the SID and writes changes back from the session cache to the global session data.

In the Response to Arguments of the Final Action, the Examiner argues that Courts teaches "the current version, previous version and the different or modified data between two versions" (Final Action, p. 16). However, Courts does not teach, even if combined with the Examiner's other cited art, does not teach the subject matter relied on by the Examiner. Presumably the Examiner contends that any reference to updating global session from a session cache includes tracking accesses of mutable attributes. However, merely writing cached session data back to global session data does include or imply tracking accesses of mutable attributes in the client state of the session data. Traditionally, accesses of mutable attributes in session data are not tracked and global session data is overwritten with the data from a session cache. Furthermore, writing data from a session cache back to the global session data in Courts only implies that the final version of data in the cache is written back to the global session data. There is nothing in Courts that teaches or suggests the tracking of mutable *accesses* of attributes. Additionally, Challenger also fails to teach tracking mutable accesses of a plurality of attributes of a client state of session data, and thus does not overcome the above noted deficiencies of Courts.

Additionally, as described above regarding claim 26, the Examiner's interpretation of Courts cannot be correct. Specifically, the Examiner refers to Courts' monitoring of client requests in support of load balancing across multiple servers to teaching tracking, while referring to Courts' session caches to teach mutable session data (See Office Action dated June 27, 2005, page 8, lines 10-12). However, Courts' monitoring of request for load balancing has nothing to do with the session caches nor about writing the session cache data back to the global session data described at the

Examiner's cited passage (column 8, lines 1-25). Thus, the Examiner contends that two separate and independent portions of Courts' system somehow teach the specific limitation of tracking mutable accesses of a plurality of attributes of a client state of session data. Such an interpretation is clearly incorrect.

The Examiner admits that Courts fails to teach performing an object graph comparison of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes, wherein the benchmark version of the client state of the session data comprises a previous version of the attributes in the client state of the session data. The Examiner relies upon Challenger and argues that Challenger teaches an object graph comparison of current and previous version of state data and cites column 10, lines 36-43 and column 28, lines 1-7. Challenger teaches the use of object dependence graphs in order to determine how different are two different caches of a single data source. In other words, Challenger is concerned with propagating changes made to a central data source to multiple data caches. (See, e.g. column 3, lines 22-27; column 9, lines 47-61; column 10, lines 14-24). While Challenger does teach using object dependence graphs for comparing different versions of data, Challenger has nothing to do with comparing mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes. Challenger in combination with Courts does not teach or suggest performing an object graph comparison *of mutably accessed attributes with a benchmark version of the client state of the session data to determine a subset of modified attributes*.

For at least the reasons presented above, the rejection of claim 29 is not supported by the prior art and removal thereof is respectfully requested. Similar remarks apply to claim 40.

Applicants also assert that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the rejection has been shown to be

unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

Allowed Claims:

Claims 1-25 have been allowed by the Examiner.

Claims Objected To But Otherwise Allowable:

Claim 28 was objected to as being dependent up on a rejected base claim, but otherwise allowable if rewritten in independent form. Applicants submit, however, that claim 28 is allowable as currently written.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-07800/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicants

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: November 13, 2007